Multimedia Tools and Applications

An Optimal Vision-based Potential Derailment Detection System

--Manuscript Draft--

Manuscript Number:	MTAP-D-19-03286
Full Title:	An Optimal Vision-based Potential Derailment Detection System
Article Type:	Manuscript
Keywords:	vision; detection; railway; tracks; buckling; hogging; learning; feature; optimal
Corresponding Author:	Koteswar Rao Jerripothula, Ph.D Graphic Era University Deharadun, INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Graphic Era University
Corresponding Author's Secondary Institution:	
First Author:	Koteswar Rao Jerripothula, Ph.D
First Author Secondary Information:	
Order of Authors:	Koteswar Rao Jerripothula, Ph.D
	Sharik Ali Ansari, B.Tech
	Rahul Nijhawan
	Ankush Mittal
Order of Authors Secondary Information:	
Funding Information:	

An Optimal Vision-based Potential Derailment Detection System

Koteswar Rao Jerripothula[2], Sharik Ali Ansari[1,3], Rahul Nijhawan[2], and Ankush Mittal[3]

[1] College of Engineering Roorkee, India
[2] Graphic Era University, India
[3] Raman Classes, India

Abstract. The majority of railway accidents happen due to derailment. Buckling and Hogging are two important track defects that can easily cause a derailment. While the buckling defect is characterized by lateral misalignment of the tracks, the hogging defect is characterized by vertical misalignment of the tracks. Therefore, these defects are visually detectable from far itself. In this paper, we present a vision-based solution to prevent railway accidents through the detection of any of such misalignments. To the best of our knowledge, there hardly any publicly available dataset to help with this problem; therefore, we introduce a new dataset named TrackDefect to help in such detections. There are numerous pre-trained networks (as feature extractors) and learning algorithms available for leveraging. We proposed an optimal predictive modeling approach over all the combinations possible for further testing and deployment, instead of just proposing one combination. In this way, we manage to claim our vision-based potential derailment system to be an optimal one. Our proposed optimal system attains about 98% test accuracy.

Keywords: Buckling · Hogging · Railways · Vision · Detection · Classification · Images.

1 Introduction

In most countries, the railway system is the cheapest and most reliable transportation system. Millions of people travel through trains everyday. Although trains are considered safer compared to other transportation modes on the ground, railway accidents do happen. Derailment is one of the most frequent ways such accidents happen. A train is said to be derailed if one or more cars or wagons run off track [7]. Derailment is a serious problem for the railway mode of transportation as they cause a large number of deaths, injuries, loss of money and public confidence in the railway system. According to the Association of American Railroads, the issues responsible for the train accidents consist of defective tracks, faulty pieces of equipment and human errors [15]. The accidents due to a faulty track contribute to a large percentage of the total accidents [15]. The rail defects due to a faulty track include corrugation or the roaring of rails, hogged rails, kinks in rails, buckling of rails, damaged rails and wear on rails. These rail defects can be divided into two categories: (i) Those which are visible from a far distance like buckling and hogging/cycle top. (ii)Those which need closer inspection like corrugation of rails, wheel burns, wear and crack on rails, rusted rails, and loose joints. In this paper, defects that are visible from far away are taken into consideration: specifically, buckling and hogging. While the buckling defect is characterized by lateral misalignment of the tracks, the hogging defect is characterized by vertical misalignment of the tracks. Although, as precautionary measures, regular manual inspections are done to identify such misalignments, still derailments do happen due to these defects. We propose a vision-based system to detect such misalignments in the tracks. It can help in automating such inspections or warning the railway drivers of such misalignments from far itself in the real-time. If such misalignments are visible, our vision-based system can output the track as defective; otherwise, it outputs the track as normal, as shown in Fig. 1. While the first track is a normal track, the second and the third have misalignments caused by buckling and hogging defects, respectively.

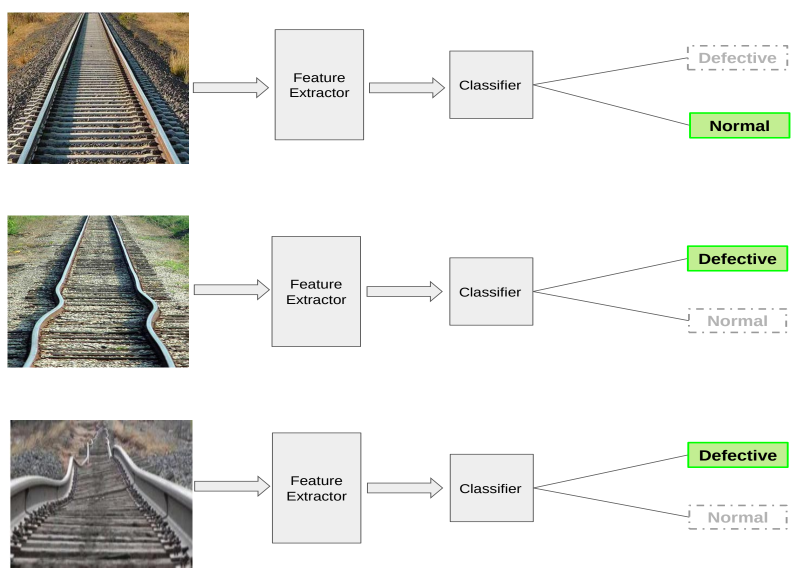

Fig. 1. Our objective is to detect defective rail-tracks. While the first track is normal, the second and the third have buckling and hogging defects, respectively.

Despite such misalignments being visible from far, it's very challenging to model them visually, for we have to segment [27, 5, 33, 14, 10] out the tracks (as done in [21]), perform shape analysis [20, 24, 12, 9], etc, which are challenging problems in the natural setting. Even if we take help of the supervised [17, 13] learning approach, the feature designing and extraction remains a big concern still, for it again has to depend upon unreliable segmentation and shape analysis. Recently, deep learning has received a lot of attention in vision research, for we can leave the job of feature engineering also with the computers only. However, it requires large amounts of data to train a network that can generate such useful features and eventually perform predictions in the hand. Interestingly, once trained in that way, we can exploit the feature extraction part of such pre-trained networks for any other problems also as long as they are somewhat related to the problem for which the pre-trained network was trained at the first place. And there are many such pre-trained networks available today to choose from. Same goes with supervised learning algorithms that can make use of these features to build a detector. Such a setup (pre-trained features + learning algorithms) at least frees us from the burden of annotating millions of images, which we would have had to do if we try to build a network from scratch for high accuracies. We can now get high accuracies even with as little as hundreds of annotated images. However, even such a dataset is not publicly available. Then, there is another obvious challenge of choosing the optimal pre-trained network and optimal supervised learning algorithm to give us an optimal classifier.

Since even hundreds of annotated images are sufficient with the pre-trained features + learning algorithms approach, we develop a dataset ourselves that contains hundreds of railway-track images with appropriate annotations of whether they are defective or normal. we name it TrackDefect dataset. Having access to such a dataset, we can now train using different predictive modeling approaches, i.e. usage of a particular pre-trained feature and particular learning algorithms, and also validate them. While both the training accuracy and validation accuracy high means less bias error, less difference between the two accuracies means less variance error. We can model these errors in terms of these two accuracies and try to minimize them together to identify an optimal approach. Once identifies, we can go ahead with testing and deployment.

Through this paper, our main contributions to the scientific community are as follows: (i) We develop a new dataset named TrackDefect dataset. (ii) We propose a novel way of arriving at an optimal predictive modeling approach from the training and validation accuracies. (iii) We are first to try out multiple feature extractors and multiple learning algorithms to solve the railway track misalignment problem comprehensively; we try out 36 combinations to be particular and show that our optimal classifier is indeed the best amongst all. The rest of the paper discusses our methodology in detail, our experiments and our conclusion.

2 Related Works

Development in technology led to the implementation various techniques to prevent train accidents. A robotic stick, proposed by [23], detects and characterizes rolling contact fatigue cracks using combined threshold and signature match algorithm. [30] proposed a 3D laser-based method for the detection of abrasion, scratch and peeling of the rail surface using K-means and decision trees. A totally different approach that uses Acoustic Emission(AE) was used in [34] to detect rail defects at high speed; they used multivariate acoustic noise cancellation and variable step-size least mean square to remove noise from waves at high speed. In [7], the detection of cracks was done using Gabor filter and segmentation based texture analysis features using AdaBoost. [28] also showed defect detection using Gabor filter. Another method of crack detection is proposed in [22] by comparing the image of rail from a dataset containing images of faulty rail track. Ultrasonic guided waves (UGW) were also utilized to detect defects by in [16]. Subsequently, the complexity of UGWs propagation between the medium of air and rail was addressed in [32], after which UGW method was able to detect 8mm cracks using 40 kHz or above frequency. Signal processing also has been explored for defect detection , such as in [8], where glassy rail diagram, neuronal network, and fuzzy logic were.

Specifically, as far as vision-based detection is concerned, [18] propose an approach where grey-level co-occurrence matrix and LBP are used to obtain texture features and then binary classification of whether the crack was present or not in rail track was done using neural network. The railway subgrade defects recognition was done using Feature Cascade, adversarial spatial dropout Network, non-maximum suppression and R-CNN in [31]. [19] performs real-time detection of rail surface defects on different speed ranging from 0.5 m/s to 6 m/s using contour of the direction chain code on morphologically processed image. [25] detects surface defects by first obtaining only the image of rail excluding other parts of the image using canny and then detecting the defect using only the image of rail using pre-trained CNN. [26, 29] tries out UAVs (Unmanned Aerial Vehicles) to solve this problem as well. Although many types of vision-based solutions have been proposed both through hand made features and deep learning, none of them try to optimize the solution as such. In this paper we have proposed a solution and optimized it as well.

3 Methodology

In this section, we discuss our methodology in detail: the way dataset was created, the way training/validation processes are performed, the way optimal predictive modeling approach is chosen, and the way final testing/deployment is performed.

3.1 Our TrackDefect Dataset

As mentioned earlier, we deal with two defects in this paper: buckling and hogging. Buckling of rails is defined as lateral misalignment of railway tracks as illustrated in Fig. 2. Buckling occurs due to the longitudinal compressive stress that builds inside the rail due to the difference in the temperature compared to the rail neutral temperature (RNT). RNT is the temperature at which the rail is in a stress-free state. The weakened condition of the track and the vehicle load also contribute to buckling, according to [11]. When the train passes through such laterally misaligned track, the train wheel may lose the track causing the train to get derailed. Reports in [2, 4] show several derailments caused due to buckling.

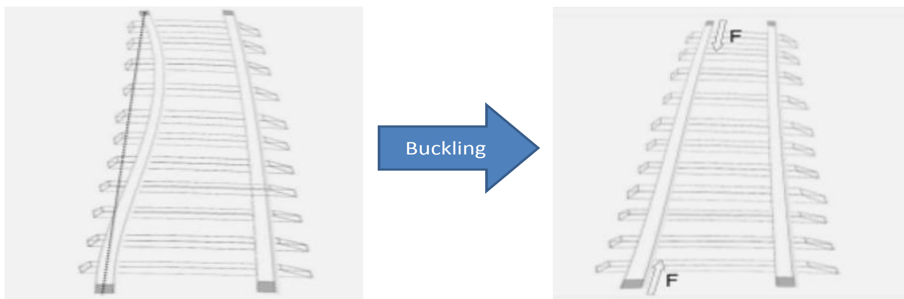

Fig. 2. Buckling Phenomenon: When there is a development of longitudinal compressive stress (represented by F), the track starts buckling from its original position (shown by a line)

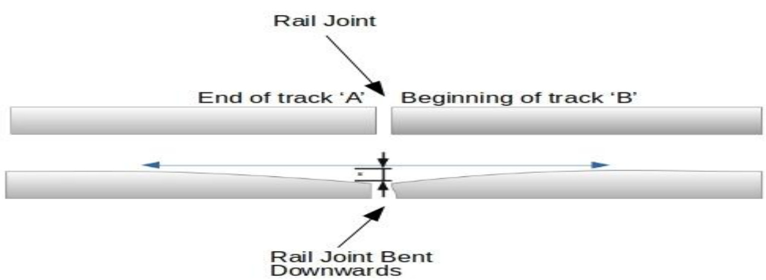

Fig. 3. Hogging Phenomenon: When there is a development of longitudinal compressive stress (represented by F), the track starts buckling from its original position (shown by a line)

Hogging of railway tracks occur primarily due to the battering (repeated hitting) action of the train wheels. It causes the tracks to bend down at their ends, where tracks are joined to form the continuity. This results in a dip (a vertical misalignment) at such joints. This phenomenon is illustrated in Fig. 3. When one such dip occurs in the railway track, the dip causes the train to bounce. And when a train bounces, it pushes the track downwards while landing. Due to such pressure, another dip starts to form as well. In this manner, a sequence of dips is created as the dip causes train to bounce [6]. The passage of many trains causes the formed dips to get deeper and deeper upto the point of derailment. Such a defect is also called cyclic-top due the formation of dips in a cyclic manner. The reports in [3, 1] show derailments occurred due to hogging/cyclic-top rail defect.

We collect several images of buckled and hogged tracks from the internet to form defective category of our TrackDefect dataset. Similarly, we also collect images of normal tracks to form normal category of our dataset. While we keep 30% of our data for testing, the remaining portion is used for training of our classifier. The exact distribution of our dataset is given in Table 1. In total, we have 435 images in our dataset with appropriate annotations of defective or normal.

Table 1. Distribution of our TrackDefect dataset

Train		Test	
Defective	Normal	Defective	Normal
107	198	46	84

3.2 Predictive Modeling Approaches

Our predictive modeling approach has two components: feature extractor (pre-trained network) and learning algorithm. There are numerous pre-trained networks that can be used as feature extractors. Let $\mathcal{F} = \{f_1, f_2, \cdots, f_m\}$ denote the set of m feature extractors we use. Similarly, There are numerous learning algorithms as well. Let $\mathcal{A} = \{a_1, a_2, \cdots, f_n\}$ denote the set of m feature extractors we use. Now, there are m (feature extractors) $\times$ n (learning algorithms) approaches possible. Out of these $m \times n$ approaches, there should be an optimal approach that strikes right balance between bias and variance errors caused while using these approaches.

3.3 Bias and Variance Error Modeling

In order to account for bias error, we need to ensure that the model is not so simplistic (or is complex). If it's so simple, it's difficult to obtain high accuracies, neither during training nor during validation. [Note: We use cross-validation strategy for validation]. So, for a given f_i and a_j, we can assume that the bias error (denoted by $B(f_i, a_j)$) is inversely proportional to the sum of training (denoted by $T(f_i, a_j)$) and validation (denoted by $V(f_i, a_j)$) accuracies:

$$B(f_i, a_j) \propto \frac{1}{T(f_i, a_j) + V(f_i, a_j) + C} \tag{1}$$

Note that we add a positive value C to have a definite error when the two accuracies are absolute 0s. Similarly, in order to account for variance error, we need to ensure that the model is generic. If it's generic, both the accuracies are required to be close. So, for a given f_i and a_j, we can assume that the variance error (denoted by $S(f_i, a_j)$) is directly proportional to the exponential of absolute difference between training and validation accuracies:

$$S(f_i, a_j) \propto e^{|T(f_i,a_j)-V(f_i,a_j)|} \tag{2}$$

Note that we vary it exponentially instead of directly. We do so to match its level of errors to that of bias error according to its formulation. Let the proportionality constants for Eqns.(1) and (2) be ρ_b and ρ_s, which will be useful in the later subsections.

3.4 Total Error

Both the errors are important and need to be considered jointly. We combine the two errors by multiplying them in the following manner to obtain our total error $\epsilon(\cdot)$:

$$\epsilon(f_i, a_j) = B(f_i, a_j) \times S(f_i, a_j) \tag{3}$$

If we substitute these errors by their models in terms of the two accuracies, we get the following:

$$\epsilon(f_i, a_j) = \frac{\rho_b}{T(f_i, a_j) + V(f_i, a_j) + C} \times \rho_s e^{|T(f_i,a_j)-V(f_i,a_j)|} \tag{4}$$

In this way, we obtain our total error. In order for both the errors to have the same contribution, we need to ensure that their ranges are same. While the range of bias error is $(\frac{\rho_b}{C+2}, \frac{\rho_b}{C})$, the range of variance error is $(\rho_s, \rho_s e)$. By equating the minimums and maximums, we can obtain the following relationships:

$$\rho_b = (C+2)\rho_s \tag{5}$$

$$\rho_s = Ce\rho_s. \tag{6}$$

By dividing the above two equations, we get $C = \frac{2}{e-1}$, which is 1.164 approximately. So, we can now write our total error as the following:

$$\epsilon(f_i, a_j) = \frac{\rho_b \rho_s}{T(f_i, a_j) + V(f_i, a_j) + 1.164} \times e^{|T(f_i,a_j)-V(f_i,a_j)|} \tag{7}$$

3.5 Objective

Our objective is to either minimize the total error or maximize its reciprocal. If we consider maximizing the reciprocal approach, Let $X(\cdot)$ be the error reciprocal as defined below:

$$X(f_i, a_j) = \Big(T(f_i, a_j) + V(f_i, a_j) + 1.164\Big) \exp^{-|T(f_i,a_j)-V(f_i,a_j)|} \tag{8}$$

Note that since $(\rho_b \rho_s)$ is just a proportionality constant multiplied to the remaining and have no role to play in maximizing, we have removed it altogether.

Now, we need to maximize the error reciprocal over different predictive modeling approaches to find the optimal approach (f_*, a_*), as shown below:

$$(f_*, a_*) = \underset{f_i \in \mathcal{F}, a_j \in \mathcal{A}}{\arg\max} X(f_i, a_j) \tag{9}$$

The above objective can be achieved through exhaustive search only. That is, we have to compute our training and validation accuracies for each combination and search for the best combination that has the maximum error reciprocal.

3.6 Testing and Deployment

Once the predictive modeling approach is finalized through the error reciprocal $X(\cdot)$, we train a model by applying it on the entire training dataset, and then, we test it on the test dataset. However, as far as deployment is concerned, since more the data better it is, we apply our optimal predictive modeling approach on the entire dataset. And to report the deployment accuracy, we apply the leave-one-out strategy on the entire dataset. The idea is absence of just one example should not alter the model learned much. So, whatever accuracy we obtain through such strategy should be equal to the actual accuracy of the deployment model.

4 Experimental Results

In this section, we give the details of experiments done and the results obtained. This section is divided into two: one is for predictive modeling approach selection, and the other is for testing and deployment results.

4.1 Predictive Modeling Approach Selection

As mentioned earlier, our predictive modeling approach consists of two components: feature extractors and learning algorithms. Specifically, we use InceptionV3, VGG16, VGG19 and SqueezeNet as possible feature extractors. And as far as learning algorithms are concerned, we use k-Nearest Neighbors (kNN), Decision Tree (DT), Support Vectors Machine (SVM), Stochastic Gradient Descent (SGD), Random Forest (RF), Neural Networks (NN), Naive Bayes (NB), Logistic Regression (LR) and AdaBoost (AB) as possibilities. So, we have total $4 \times 9 = 36$ approaches possible, and we need to select the optimal one according to Eqn.(9). The training and validation accuracies of each approach are given in Tables 3 and 2, respectively. By accuracy, we mean classification accuracy. From these two tables, we obtain our error reciprocal $X(\cdot)$ for each approach in Table 4. It can be observed from the table that amongst all the 36 approach, VGG16+NN approach gives the best error reciprocal value, i.e., 1.5027. Therefore, we consider VGG16+NN approach as the required optimal predictive modeling approach.

Table 2. Cross Validation accuracies of different predictive modeling approaches

	SqueezeNet	**Inception v3**	**VGG 16**	**VGG 19**
kNN	0.911	0.898	0.921	0.905
DT	0.836	0.846	0.862	0.826
SVM	0.928	0.925	0.921	0.915
SGD	0.905	0.918	0.944	0.921
RF	0.908	0.872	0.892	0.892
NN	0.931	0.941	0.961	0.951
NB	0.872	0.813	0.875	0.846
LR	0.915	0.928	0.954	0.941
AB	0.843	0.839	0.836	0.800

Table 3. Training accuracies of different predictive modeling approaches

	SqueezeNet	**Inception v3**	**VGG 16**	**VGG 19**
kNN	0.944	0.921	0.957	0.954
DT	0.993	0.993	0.987	0.987
SVM	0.970	0.993	0.990	0.974
SGD	1.000	1.000	1.000	1.000
RF	1.000	0.997	0.997	0.990
NN	1.000	1.000	1.000	1.000
NB	0.895	0.869	0.898	0.879
LR	1.000	1.000	1.000	1.000
AB	1.000	1.000	1.000	1.000

Table 4. Our error reciprocal $X(\cdot)$ of different predictive modeling approaches

	SqueezeNet	**Inception v3**	**VGG 16**	**VGG 19**
kNN	1.461	1.458	1.467	1.439
DT	1.279	1.296	1.33	1.267
SVM	1.468	1.44	1.435	1.439
SGD	1.395	1.42	1.469	1.425
RF	1.401	1.338	1.374	1.381
NN	1.444	1.464	**1.503**	1.483
NB	1.432	1.346	1.435	1.398
LR	1.414	1.439	1.489	1.464
AB	1.285	1.278	1.273	1.213

4.2 Testing and Deployment

Having obtained the optimal predictive modeling approach as VGG16+NN, we compare its test accuracy on our testing dataset with all other approaches in Table 5. It can be seen that our optimal approach (VGG16+NN) indeed gives the best test accuracy amongst all the possible approaches. So, we did find an optimal approach using our formulation of error reciprocal $X(\cdot)$. We also compare our optimal approach with other approaches that use the same feature extractor in terms of different evaluation metrics of classification in Table 6. So, it's not just the single metric in which our optimal approach performs the best but in others as well.

Table 5. Testing accuracies of different predictive modeling approaches

	SqueezeNet	**Inceptionv3**	**VGG16**	**VGG19**
kNN	0.931	0.915	0.954	0.931
DT	0.846	0.854	0.885	0.877
SVM	0.962	0.923	0.931	0.908
SGD	0.923	0.938	0.954	0.938
RF	0.931	0.885	0.923	0.9
NN	0.954	0.946	**0.977**	0.938
NB	0.923	0.877	0.915	0.892
LR	0.931	0.923	0.954	0.962
AB	0.792	0.869	0.846	0.85

Table 6. Comparison of optimal approach with other approaches that use the same feature in terms of different evaluation metrics

	AUC	**CA**	**F1**	**Precision**	**Recall**
kNN	0.993	0.954	0.953	0.955	0.954
Tree	0.856	0.885	0.885	0.887	0.885
SVM	0.985	0.931	0.931	0.931	0.931
SGD	0.954	0.954	0.954	0.955	0.954
Random Forest	0.974	0.923	0.923	0.923	0.923
Neural Network	**0.996**	**0.977**	**0.977**	**0.977**	**0.977**
Naive Bayes	0.935	0.915	0.916	0.916	0.915
Logistic Regression	0.99	0.954	0.954	0.954	0.954
AdaBoost	0.817	0.846	0.843	0.845	0.846

We also report the confusion matrix of the optimal approach on our test dataset. It can be seen that only 3 out of 130 images have got misclassified, and the rest are correctly classified. In Figs. 4 and 5, we give sample images that have been correctly classified as defective and normal, respectively. It's clear that our classifier that is built using the optimal predictive modeling approach is able to classify the images into defective and normal quite well. However, there were 3 images for which it couldn't. We also give those 3 in Fig. 6. One type of misclassification happened when our classifier got confused between whether it is a track joining or the buckling defect because both can look similar at times. The other type happened when the defect is not so acute: It still looks almost like a proper track. These problems can be eliminated if we can have more data for learning in order to discriminate such fine details.

Table 7. Confusion Matrix of our approach (VGG16+NN) on the test dataset

	Predicted Defective	Predicted Normal
Actually Defective	45	1
Actually Normal	2	82

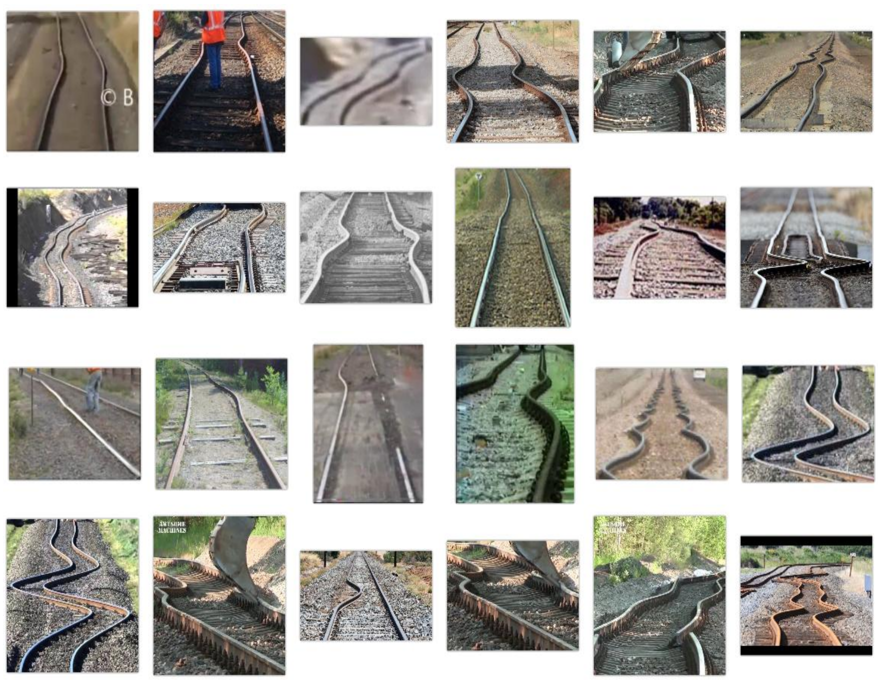

Fig. 4. Sample images that have been classified as defective by our classifier correctly.

As far as deployment is concerned, we can employ the whole dataset for learning our deployment model. In order to evaluate how good it is, we employ Leave-one-out approach on the entire dataset. When we just leave one out, the absence of just one example may not change deployment model much, but at the

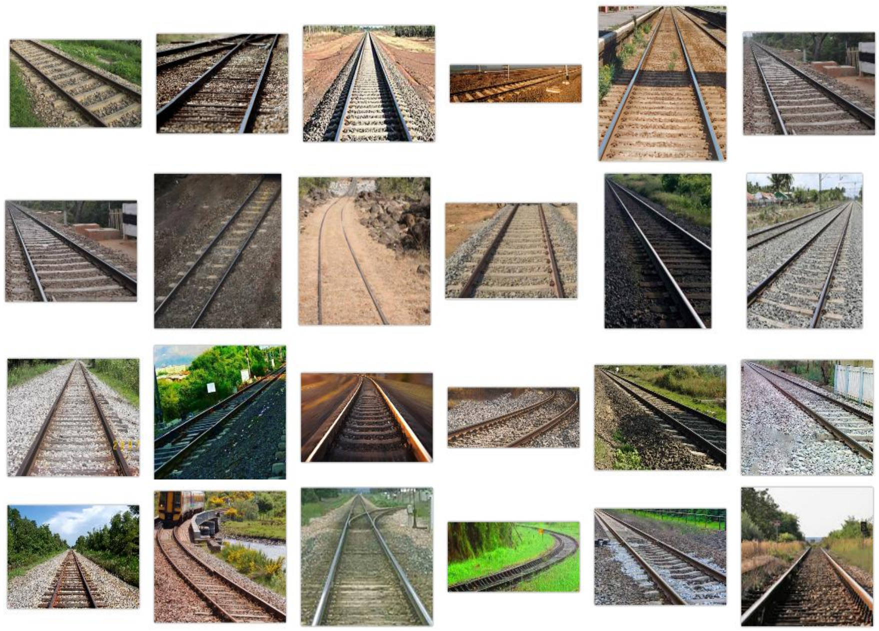

Fig. 5. Sample images that have been classified as normal by our classifier correctly.

Fig. 6. Sample images for which our classifier fails to classify appropriately.

same time it allows us to test on an unseen data. This leave one out accuracy of our deployment model is given in Table 8 in terms of various evaluation metrics.

Table 8. Deployment Accuracy: LOO acuracy on the entire Dataset

Evaluation Metrics	**AUC**	**CA**	**F1**	**Precision**	**Recall**
Values	0.989	0.966	0.965	0.965	0.966

Conclusion

We develop a vision-based potential derailment detection system. We develop a dataset with images of tracks having defects like buckling and hogging and the images of tracks that are normal. We name the dataset as TrackDefect. We extract several features using pre-trained networks and apply different learning algorithms to search for an optimal approach (we don't know which feature extractor and which learning algorithm to use). We develop a novel metric called error reciprocal by modeling the bias and variance errors in terms of training and validation accuracies. Whichever approach yields the maximum error reciprocal value, we choose that particular approach as optimal one. We found that to be VGG16+NN in our case. For the same approache, we even obtain best test accuracy, which suggests that the proposed accuracy does perform the job of selecting the optimal approach. We obtain 97.7% test accuracy on the test dataset. Also, we obtain 96.6% deployment accuracy using leave-one-out approach on the entire dataset.

References

1. Derailment of train 4vm9-vbenalla, victoria. Tech. rep., Australian Transport safety Bureau (2004)
2. Railway investigation report r05h0013. Tech. rep., Transportation Safety Board of Canadan (2005)
3. Report 20/2014: Freight train derailment near gloucester. Tech. rep., Rail Accident Investigation Branch, Department for Transport, UK (2014)
4. Report 11/2016: Derailment near langworth. Tech. rep., Rail Accident Investigation Branch, Department for Transport, UK (2016)
5. Alsmirat, M.A., Jararweh, Y., Al-Ayyoub, M., Shehab, M.A., Gupta, B.B.: Accelerating compute intensive medical imaging segmentation algorithms using hybrid cpu-gpu implementations. Multimedia Tools and Applications **76**(3), 3537–3555 (2017)
6. apheby: Cyclic top. https://safety.networkrail.co.uk/jargon-buster/cyclic-top/ (2016), [Online; accessed 14-Sept-2019]
7. Arivazhagan, S., Shebiah, R.N., Magdalene, J.S., Sushmitha, G.: Railway track derailment inspection system using segmentation based fractal texture analysis. ICTACT Journal on Image & Video Processing **6**(1) (2015)

8. Heckel, T., Casperson, R., Rühe, S., Mook, G.: Signal processing for non-destructive testing of railway tracks. In: AIP Conference Proceedings. vol. 1949, p. 030005. AIP Publishing (2018)
9. Jerripothula, K.R., Cai, J., Lu, J., Yuan, J.: Object co-skeletonization with co-segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3881–3889 (July 2017). https://doi.org/10.1109/CVPR.2017.413
10. Jerripothula, K.R., Cai, J., Yuan, J.: Quality-guided fusion-based co-saliency estimation for image co-segmentation and colocalization. IEEE Transactions on Multimedia **20**(9), 2466–2477 (Sep 2018). https://doi.org/10.1109/TMM.2018.2798294
11. Kish, A., Mui, W.: Track buckling research report. Tech. rep., United States Department of Transportation (2003)
12. Li, B., Godil, A., Johan, H.: Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3d model retrieval. Multimedia tools and applications **72**(2), 1531–1560 (2014)
13. Liu, A.A., Shi, Y., Nie, W.Z., Su, Y.T.: View-based 3d model retrieval via supervised multi-view feature learning. Multimedia Tools and Applications **77**(3), 3229–3243 (2018)
14. Liu, L., Zhang, J., Fu, X., Liu, L., Huang, Q.: Unsupervised segmentation and elm for fabric defect image classification. Multimedia Tools and Applications **78**(9), 12421–12449 (2019)
15. Liu, X., Saat, M.R., Barkan, C.P.: Analysis of causes of major train derailment and their effect on accident rates. Transportation Research Record **2289**(1), 154–163 (2012)
16. Loveday, P.W., Taylor, R.M., Long, C.S., Ramatlo, D.A.: Monitoring the reflection from an artificial defect in rail track using guided wave ultrasound. In: AIP Conference Proceedings. vol. 1949, p. 090003. AIP Publishing (2018)
17. Makula, P., Kumar, A., Mukherjee, S.: Measuring level of cuteness of baby images: a supervised learning scheme. Multimedia Tools and Applications **77**(13), 16867–16885 (2018)
18. Manikandan, R., Balasubramanian, M., Palanivel, S.: An efficient framework to detect cracks in rail tracks using neural network classifier. International Journal of Pure and Applied Mathematics **116**(10), 461–470 (2017)
19. Min, Y., Xiao, B., Dang, J., Yue, B., Cheng, T.: Real time detection system for rail surface defects based on machine vision. EURASIP Journal on Image and Video Processing **2018**(1), 3 (2018)
20. Mokni, R., Drira, H., Kherallah, M.: Combining shape analysis and texture pattern for palmprint identification. Multimedia Tools and Applications **76**(22), 23981–24008 (2017)
21. Ritika, S., Rao, D.: Data augmentation of railway images for track inspection. CoRR **abs/1802.01286** (2018), http://arxiv.org/abs/1802.01286
22. Rizvi, A.R., Khan, P.R., Ahmad, S.: Crack detection in railway track using image processing. International Journal of Advance Research, Ideas and Innovations in Technology **3**(4), 489–496 (2017)
23. Rowshandel, H.: The development of an autonomous robotic inspection system to detect and characterise rolling contact fatigue cracks in railway track. Ph.D. thesis, University of Birmingham (2014)
24. Shahabi, C., Safar, M.: An experimental study of alternative shape-based image retrieval techniques. Multimedia Tools and Applications **32**(1), 29–48 (2007)
25. Shang, L., Yang, Q., Wang, J., Li, S., Lei, W.: Detection of rail surface defects based on cnn image recognition and classification. In: 2018 20th International Conference on Advanced Communication Technology (ICACT). pp. 45–51. IEEE (2018)

26. Singh, A.K., Swarup, A., Agarwal, A., Singh, D.: Vision based rail track extraction and monitoring through drone imagery. ICT Express (2017)
27. Teng, Z., Liu, F., Zhang, B.: Visual railway detection by superpixel based intracellular decisions. Multimedia Tools and Applications **75**(5), 2473–2486 (2016)
28. Vijaykumar, V., Sangamithirai, S.: Rail defect detection using gabor filters with texture analysis. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN). pp. 1–6. IEEE (2015)
29. Wu, Y., Qin, Y., Wang, Z., Jia, L.: A uav-based visual inspection method for rail surface defects. Applied Sciences **8**(7), 1028 (2018)
30. Xiong, Z., Li, Q., Mao, Q., Zou, Q.: A 3d laser profiling system for rail surface defect detection. Sensors **17**(8), 1791 (2017)
31. Xu, X., Lei, Y., Yang, F.: Railway subgrade defect automatic recognition method based on improved faster r-cnn. Scientific Programming **2018** (2018)
32. Yao, W., Yang, Y., Zhao, K., Wei, X., Zhang, L., Zhai, M., Rao, F.: Rail defect inspection based on coupled structural–acoustic analysis. International Journal of Modern Physics B **33**(01n03), 1940031 (2019)
33. Yu, H., He, F., Pan, Y.: A novel region-based active contour model via local patch similarity measure for image segmentation. Multimedia Tools and Applications **77**(18), 24097–24119 (2018)
34. Zhang, X., Cui, Y., Wang, Y., Sun, M., Hu, H.: An improved ae detection method of rail defect based on multi-level anc with vss-lms. Mechanical Systems and Signal Processing **99**, 420–433 (2018)